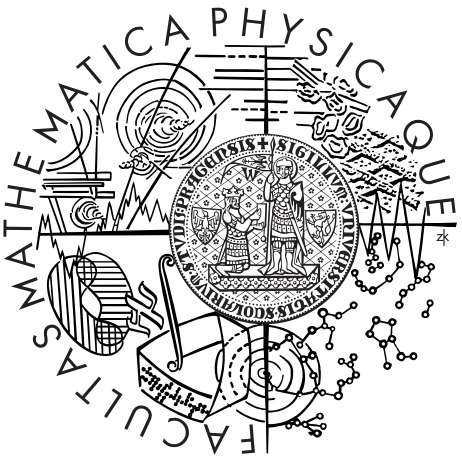
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Eduard Ješko

Kvantové počítače, jejich principy a nedávný vývoj

Katedra chemické fyziky a optiky

Vedoucí bakalářské práce: prof. RNDr. Lubomír Skála, DrSc.

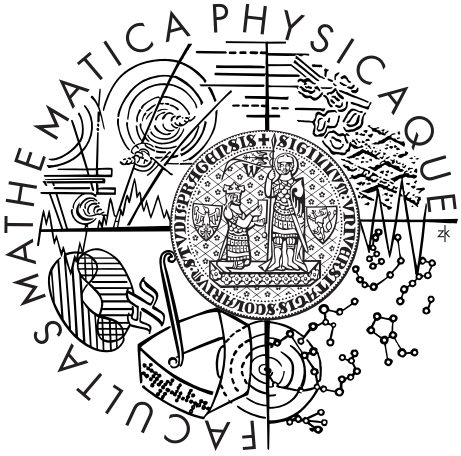
Studijní program: Fyzika

Studijní obor: Obecná fyzika

Praha 2014

Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS



Eduard Ješko

Quantum computers, principles and latest development

Department of Chemical Physics and Optics

Supervisor of the bachelor thesis: prof. RNDr. Lubomír Skála, DrSc.

Study programme: Physics

Specialization: General Physics

Prague 2014

Acknowledgment

I would like to express my deepest gratitude to my supervisor prof. RNDr. Lubomír Skála, DrSc. and consultant Mgr. Jiří Pittner, Dr. for their leadership, patience and willingness. They offered invaluable assistance and guidance. Also, I would like to thank my parents for their unconditional love and support throughout my studies.

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, date 23.5.2014

Název práce: Kvantové počítače, jejich principy a nedávný vývoj

Autor: Eduard Ješko

Katedra: Katedra chemické fyziky a optiky

Vedoucí bakalářské práce: prof. RNDr. Lubomír Skála, DrSc., Katedra chemické fyziky a optiky

Abstrakt: Cieľom tejto práce je podať základné informácie o princípoch kvantových počítačov a ich nedávneho vývoja. V prvej časti práce sú vysvetlené pojmy kvantový bit, kvantový register a kvantové hradlo. Predstavené sú základné operácie pôsobiace na jeden a viacero qubitov a na príklade je ukázané ako pomocou základných hradiel zostrojiť ľubovoľné hradlo. V ďalšej časti je popísané fungovanie kvantového paralelizmu a jeho aplikácia v Deutschovom algoritme. Zavádza sa kvantová Fourierovú transformáciu a jej použitie. V poslednej kapitole je objasnené fungovanie kvantového počítača zostrojeného spoločnosťou D-Wave. Práca kladie dôraz na porovnávanie algoritmov a rýchlosti výpočtu klasického a kvantového počítača.

Klíčová slova: kvantové počítače, principy, nedávný vývoj

Title: Quantum computers, principles and latest development

Author: Eduard Ješko

Department: Department of Chemical Physics and Optics

Supervisor of the bachelor thesis: prof. RNDr. Lubomír Skála, DrSc., Department of Chemical Physics and Optics

Abstract: The main goal of this thesis is to give information on quantum computer principles and its latest development. In the first part we introduce quantum bits, quantum registers and quantum gates. We show basic operations acting on one and more qubits. On an example we present, how it is possible to construct an arbitrary gate using only elementary quantum gates. We describe a behaviour of quantum computers called quantum parallelism and show its application in Deutsch's algorithm. We define the quantum Fourier transform and its applications. In the last chapter we explain on what principle the D-Wave quantum computer works. In this thesis we compare classical and quantum computers in terms of algorithms and computational speed.

Keywords: quantum computers, principles, latest development

Contents

1 Motivation and purpose of Quantum Computers 2
1.1 Development of classical computers . . . 2
1.2 Quantum computers and quantum computing . . . 3

2 Basics of Quantum Computing 5
2.1 Quantum bits . . . 5
2.1.1 Classical bit vs. qubit . . . 5
2.1.2 Two and more qubit systems . . . 6
2.1.3 Measuring of qubits . . . 7
2.2 Quantum gates . . . 7
2.2.1 Single qubit gates . . . 8
2.2.2 Controlled gates . . . 11

3 Quantum algorithms 16
3.1 Quantum parallelism . . . 16
3.2 Deutsch's algorithm . . . 18
3.3 Quantum Fourier transform . . . 20
3.3.1 Phase estimation . . . 22
3.3.2 Order-finding . . . 23
3.3.3 Factoring . . . 24

4 Latest development of quantum computers 26
4.1 Superconducting flux qubits . . . 27
4.2 Quantum annealing and adiabatic quantum computation . . . 29
4.2.1 Annealing . . . 29
4.2.2 Adiabatic quantum computation . . . 29
4.3 Experiment on the D-Wave One computer . . . 30

Summary and Conclusion 33

Bibliography 34

List of Figures 36

List of Tables 37

Chapter 1

Motivation and purpose of Quantum Computers

Nothing is impossible to a willing heart.
— English writer John Heywood

1.1 Development of classical computers

Intensive research of classical computers began during World War II, when there was a need to calculate with large numbers and difficult problems in the Manhattan project. Many scientists were brought together to face this problem. The first computers they built were so big that they filled up a whole room, which restricted their wider application. Fortunately in the late 1940s transistors were invented by William Shockley, John Bardeen and Walter Brattain, which lead to massive development of classical computers.

In the next decades the computational power has risen fast. This increase was examined by Gordon E. Moore who observed that the number of transistors doubles roughly every two years. The capabilities of many electronic devices are strongly related to this law: memory capacity, processing speed or sensors. This exponential increase has a huge impact in every segment of the world economy. The increase of computational power and decrease of size of smartphones, tablets or laptops is astonishing. The dependence of the number of transistors on time is shown in Figure 1.1.

However this exponential increase has its boundaries which will show according to experts at the end of 2015. The increase in performance is due to the fact that we lay more transistors on a same size chip. That is why large processor manufacturers such as Intel or AMD try to produce smaller and smaller transistors. However when the distance between two transistors will get smaller then 10^{-9}m disturbing quantum mechanical effects will take place and the transistors will no longer work properly. The second factor that speaks against smaller transistors is protection from overheating. At these

small distances it is almost impossible to cool the transistors by air. Liquid cooling would be expensive for commercial use.

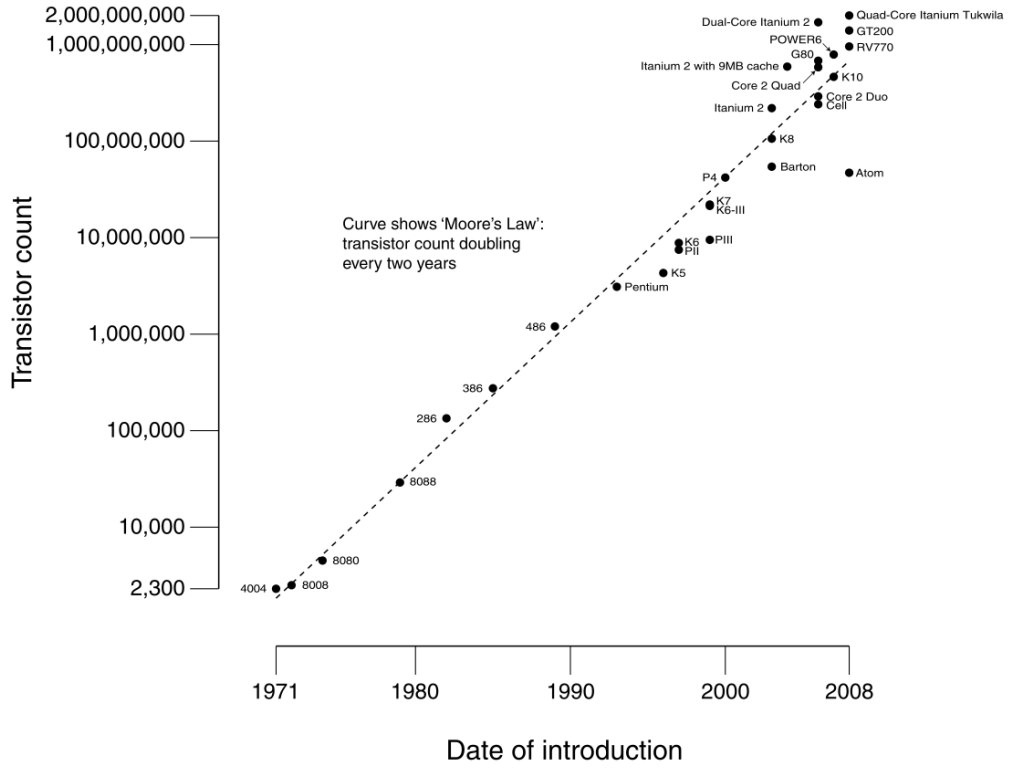


Figure 1.1: CPU transistor counts against dates of introduction. Note the logarithmic vertical scale [7].

1.2 Quantum computers and quantum computing

What should we do next? How can we increase the power of classical computers? The answer may lie in the field of quantum mechanics. Richard Feynman proposed, that instead of classical computers, which are working under the laws of classical physics, we could exploit the richness of quantum mechanics. In quantum computers bits are replaced by quantum bits (called *qubits*) on which the computation is performed. At the moment there are already existing quantum computers which are able to run special algorithms. We will discuss the construction and experimental aspects of quantum computing in Chapter 4.

Let us consider we constructed a quantum computer. We can ask ourselves a question. Will there be a difference between classical software and quantum software? The answer is yes. Since quantum mechanics is a generalization of

classical mechanics, we should be able to construct algorithms which will be more efficient then their classical counterparts. Several quantum algorithms will be described in Chapter 3. Quantum mechanics opens us a new approach to computation and we will try to explain the basics in the following chapters.

Chapter 2

Basics of Quantum Computing

2.1 Quantum bits

2.1.1 Classical bit vs. qubit

As we mentioned earlier, *bits* are the basic units of information in computing. It is well known that bits can gain only two different values, either 0 or 1, but only one at the same time. Will qubits have similar properties? We will try to answer this question in this chapter.

First we will define the two *computational basis vectors* $|0\rangle$ and $|1\rangle$

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

These basis vectors correspond to the classical bit values 0 and 1. The main difference between qubits and classical bits is, that qubits can be in a superposition of these two states, that is

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ are the so called *probability amplitudes*. These two probability amplitudes must satisfy the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.2}$$

We can rewrite equation (2.1) in a different form, which will help us visualize a single qubit

$$|\psi\rangle = e^{i\delta}\left(\cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle\right), \tag{2.3}$$

where $\theta, \varphi, \delta \in \mathbb{R}$. Since the factor $e^{i\delta}$ in Equation 2.3 has no observable effect, it will be omitted.

Geometrically we can represent the state of a single qubit (described by Equation (2.3)), without the global phase on the *Bloch sphere*[1], shown in Figure 2.1. Any point on the Bloch sphere will be defined by numbers θ and φ.

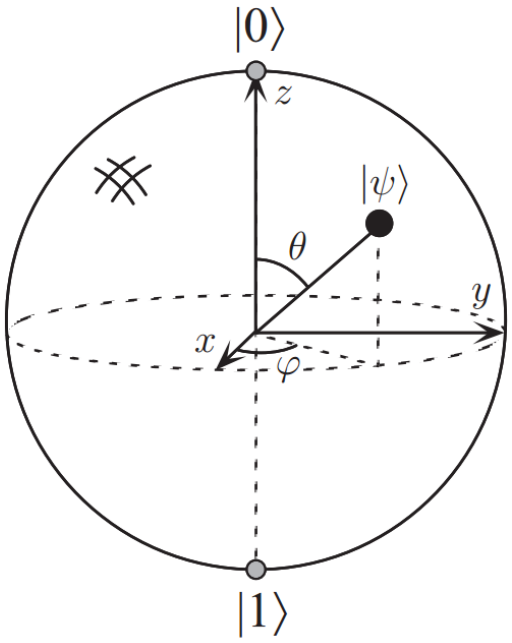


Figure 2.1: Bloch sphere [8].

2.1.2 Two and more qubit systems

Naturally, there is a need for more qubits to perform more advanced computation. A collection of n qubits is called a *quantum register*. The state of a quantum register is expressed by the tensor product of the states of each qubit, that is

$$|\psi\rangle = |qubit_{N-1}\rangle \otimes |qubit_{N-2}\rangle \otimes \ldots \otimes |qubit_1\rangle \otimes |qubit_0\rangle. \tag{2.4}$$

It may contain any of the $N = 2^n$-dimensional computational basis vectors, n qubit of size, or arbitrary superposition of these vectors.

We can show on a simple example with two qubits

$$|\psi_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, |\psi_2\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Now we join the two qubits into a quantum register $|\psi\rangle$

[1] Named after Swiss physicist Felix Bloch.

$$|\psi\rangle \equiv |\psi_1\rangle|\psi_2\rangle \equiv |\psi_1\psi_2\rangle = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle}{2}$$
$$= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}.$$

We have shown, that the state of a two qubit register consists of four, linearly weighted computational basis vectors. We got 4 new vectors: $|00\rangle, |10\rangle, |01\rangle$ and $|11\rangle$ which are the potential contents of a classical two-bit register. In our quantum case, we observe, that all of them are only in a single quantum register.

Suppose we have a n qubit register. Then its general state can be characetrized by

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \tag{2.5}$$

where $\alpha_k \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. This means that such a quantum register contains 2^n different probability amplitudes (numbers)[2] at the same time!

2.1.3 Measuring of qubits

Our main goal is to gain information from the qubits. That means we will have to measure them. The third postulate of quantum mechanics says that *any measurement of the observable* Γ *associated with operator* $\hat{\Gamma}$*, will convert the measured system into its eigenstate*. In our case this means that before the measurement the qubit has both logical values, but after the measurement we will obtain $|0\rangle$ with the probability α^2 and $|1\rangle$ with the probabilty β^2. For example, a qubit can be in a state

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \tag{2.6}$$

which when measured many times, gives the result $|0\rangle$ in one half of the cases and $|1\rangle$ also in one half of the cases.

2.2 Quantum gates

We will want to perform operations on qubits. That means, we will have to send a qubit through a logical gate. We will divide quantum gates into:

[2] If $n = 500$, then this number is larger then the estimated number of atoms in the Universe.

- Single qubit gates
- Multiple qubit gates

We can ask ourselves a question. Will there be any requirement on quantum gates?

Every quantum state must fulfill Equation (2.2). After applying a gate on state, the new state $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$ has to fulfill the same normalization condition. It can be shown that the matrix describing the qubit gate has to be *unitary*[3].

2.2.1 Single qubit gates

Gates acting on a single qubit will be represented by matrices size 2×2. In the following sections we will show the matrix representations of the quantum gates and next to them its circuit representation. Most important single qubit gates are:

- Pauli X gate
- Pauli Y gate
- Pauli Z gate
- Phase shift gate
- Hadamard gate
- Phase gate

Pauli X, Y, Z gates

Pauli gates are useful, mainly because they can rotate vectors. For example, the Pauli X gate rotates a state represented by a vector on a Bloch sphere around the X-axis by π radians. It is also the quantum equivalent of the NOT gate: it turns $|0\rangle$ into $|1\rangle$ and vice versa. It is represented by the Pauli X matrix

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{—}\boxed{X}\text{—} \tag{2.7}$$

The Pauli Y gate rotates a vector around the Y-axis of the Bloch sphere by π radians. It turns state $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ into $-i|0\rangle$. It is represented by the Pauli Y matrix

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{—}\boxed{Y}\text{—} \tag{2.8}$$

[3] A complex square matrix U is unitary if $U^\dagger U = UU^\dagger = I$.

The Pauli Z gate rotates a vector around the Z-axis of the Bloch sphere by π radians. It does not change on state $|0\rangle$ and it turns $|1\rangle$ into $-|1\rangle$. It is represented by the Pauli Z matrix

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{—}\boxed{Z}\text{—} \tag{2.9}$$

Phase shift gate

The phase shift gate does not change state $|0\rangle$ and changes $|1\rangle$ to $e^{i\varphi}|1\rangle$. This gate does not change the probability of measuring states $|0\rangle$ and $|1\rangle$, however it modifies the phase of state $|1\rangle$. This is equivalent to tracing a horizontal circle (a line of latitude) on the Bloch sphere by φ radians. For $\varphi = \pi$ we get the Pauli Z gate. It is represented by the matrix

$$\varphi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \quad \text{—}\boxed{\varphi}\text{—} \tag{2.10}$$

An important case is if $\varphi = \pi/4$. This gate is called the $\pi/8$ *gate*[4] which will be essential for building a universal quantum gate. Its matrix representation are

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad \text{—}\boxed{T}\text{—} \tag{2.11}$$

The Pauli matrices are a special case of the *rotation operators* (these are also unitary). The *rotation operators* about the x, y and z axes are defined by equations:

$$R_x(\theta) \equiv e^{-i\theta X/2} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right) X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.12}$$

$$R_y(\theta) \equiv e^{-i\theta Y/2} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right) Y = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.13}$$

$$R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right) Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{2.14}$$

With the help of these rotation matrices and phase shift gate we can create an arbitrary unitary operator on a single qubit. We will show a theorem called *Z-Y decomposition for a single qubit*, with the help of which we will express an arbitrary single qubit rotation.

[4] This gate is called $\pi/8$ not $\pi/4$ due to historical reasons.

Theorem 1 (Z-Y decomposition for a single qubit). *Suppose U is a unitary operation on a single qubit. Then there exist real numbers α, β, γ and δ such that*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) \tag{2.15}$$

Proof. Substitute expressions (2.13) and (2.14) into equation (2.16).

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)}\cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)}\sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)}\sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)}\cos\frac{\gamma}{2} \end{bmatrix} \tag{2.16}$$

□

Corollary. Suppose U is a unitary gate on a single qubit. Then there exist unitary operators A, B, C, where $A \equiv R_z(\beta)R_y(\gamma/2), B \equiv R_y(-\gamma/2)R_z(-(\delta+\beta)/2)$ and $C \equiv R_z((\delta-\beta)/2)$ on a single qubit that $ABC = I$ and $U = e^{i\alpha}AXBXC$, where α is some overall phase factor.

This theorem and its corollary will come useful in Section 2.2.2 when constructing controlled gates. The proof of this corollary can be found in [8].

Hadamard gate

The Hadamard gate is one of the most important gates in quantum computing. It turns the basis state $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle$ into $(|0\rangle - |1\rangle)/\sqrt{2}$. These two states are remarkable, because they represent non basis states, in which $|0\rangle$ and $|1\rangle$ can occur with the same probability. Geometrically this gate represents the rotation of π about the y axis. It is represented by the matrix

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad —\boxed{H}— \tag{2.17}$$

Phase gate

The phase gate is an important element for building a universal quantum gate. For now we will introduce only its circuit and matrix form and later in Section 2.2.2 we will show its importance by constructing an arbitrary quantum gate.

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad —\boxed{S}— \tag{2.18}$$

Universality of the Hadamard and the phase shift gate

It can be shown, that by acting two Hadamard and two phase shift (slightly modified, see proof) gates in the correct order on state $|0\rangle$, we can generate arbitrary state of a qubit [2]

$$|0\rangle \rightarrow \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle. \tag{2.19}$$

Proof.

$$\begin{bmatrix}1 & 0\\ 0 & e^{i(\frac{\pi}{2}+\varphi)}\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\theta}{2}}\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} =$$

$$= \frac{1}{2}\begin{bmatrix}1+e^{i\theta}\\ e^{\frac{\pi}{2}+\varphi}(1-e^{i\theta})\end{bmatrix} = e^{i\frac{\theta}{2}}\begin{bmatrix}\cos\frac{\theta}{2}\\ e^{\varphi}\sin\frac{\theta}{2}\end{bmatrix}$$

Since the global phase factor can be omitted (shown in section 2.1.1), the Hadamard and the phase shift gate can represent an arbitrary quantum gate.

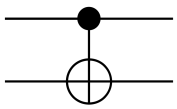
2.2.2 Controlled gates

A very important part of quantum computing is to perform operation on multiple qubit systems. For example in the case of two qubits: If the first qubit is in state $|1\rangle$ then we change the state of the second qubit, else do nothing. These kind of operations are represented by *controlled gates*.

Controlled-NOT gate

One of the most important controlled gates is the controlled-NOT (later we will refer to it as CNOT) gate. The input of this gate are two qubits, called the *control qubit* and the *target qubit*. It will perform the following operation: if the control qubit is in state $|1\rangle$, then the target qubit is flipped, otherwise the target qubit is left unchanged. The circuit and matrix representation of the CNOT gate is (the top line represents the control qubit, the bottom line the target qubit)

$$C = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \qquad (2.20)$$



For better understanding of the CNOT gate we attached the truth table of the CNOT gate in Table 2.1.

Table 2.1: Truth table of the CNOT gate ($\oplus$ stands for addition modulo two).

INPUT		OUTPUT	
x	y	x	$y \oplus x$
0	0	0	$0 \oplus 0 = 0$
0	1	0	$1 \oplus 0 = 1$
1	0	1	$0 \oplus 1 = 1$
1	1	1	$1 \oplus 1 = 0$

Controlled-U gate and the implementation of controlled operation using only single qubit operations

For the following purposes let us consider an arbitrary single qubit operation U. This unitary operation will act in a similar way as the CNOT gate. That is if the control qubit is in state $|1\rangle$, then we will apply U on the target qubit, otherwise nothing is done. We will call this the controlled-U operation and it will be represented by the circuit shown in Figure 2.2.

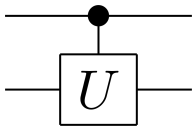


Figure 2.2: Controlled U operation.

Finally we can use our knowledge gained in the previous sections and show how to implement the controlled-U operation for arbitrary single qubit operation U, using only single qubit operations and the CNOT gate.

First we will act on the target qubit with the phase shift gate if the control qubit is $|1\rangle$. The corresponding single qubit operation is shown on the right side of Figure 2.3.

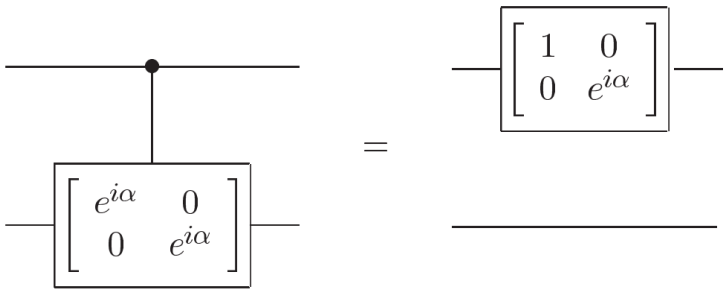


Figure 2.3: Controlled phase shift gate on the left side as a two qubit and on the right side as a single qubit operation [8].

In the second step we will use Corollary of Theorem 1, that is, $U = e^{i\alpha}AXBXC$, where A, B and C are single qubit operations and $ABC = I$. Now we can easily see, that if the control qubit is $|1\rangle$ then U is applied, otherwise $ABC = I$ is applied on the target qubit (in other words nothing is done). The final controlled-U operation is shown in Figure 2.4.

Now let us complete this section by generalizing the controlled-U operation on a set of $n + l$ qubits, where n is the number of control qubits and l is the number of target qubits. Suppose U is a unitary gate that acts on the rest l qubits. We define $C^n(U)$ operation:

$$C^n(U)|x_1x_2\ldots x_n\rangle|\psi\rangle = |x_1x_2\ldots x_n\rangle U^{x_1x_2\ldots x_n}|\psi\rangle, \tag{2.21}$$

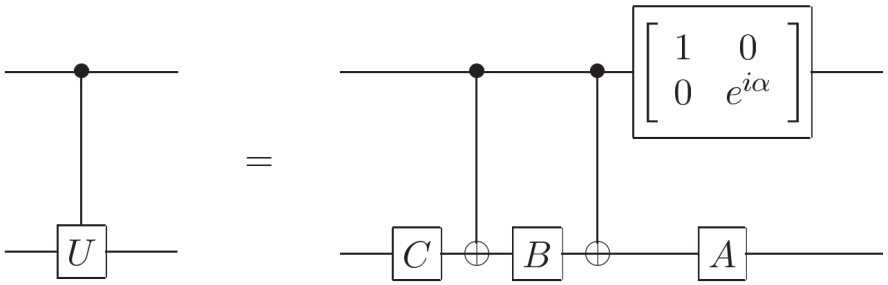


Figure 2.4: Controlled phase shift gate and equivalent circuit for two qubits [8].

where $x_1x_2 \ldots x_n$ in the exponent of U means the product of the bits $x_1, x_2, \ldots, x_n$. This multi qubit operator works on the same principle as the two qubit operator. That is, if the n control qubits are $|1\rangle$, then U acts on the l target qubits, else nothing is done.

Two more important control operations will be shown. The *Toffoli gate* and the *Fredkin gate*[5]. These two gates are used also by classical computers and since they are reversible, we can use them as quantum gates.

Fredkin gate

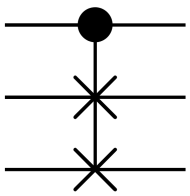
Before introducing the Fredkin gate, it is useful to show the SWAP gate. It is a two qubit gate which does nothing else but swaps two qubits. Its matrix and circuit representation

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.22)$$

The Fredkin gate (also called the controlled swap gate) is a three qubit gate which performs the following operation. If the control qubit is $|1\rangle$, then the two target qubits are swapped. If the control qubit is $|0\rangle$, then the two target qubits are left alone. For better understanding we attached the truth values of the Fredkin gate in Table 2.2.

[5]Tommaso Toffoli is an Italian electrical and computer engineer and Edward Fredkin is an American digital physics pioneer.

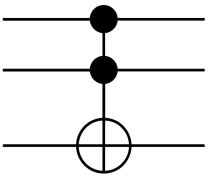
$$\text{Fredkin} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.23)$$



Toffoli gate

The Toffoli gate, often called CCNOT gate is a three qubit gate with two control qubits and one target qubit. If the two control qubits are set, then the target qubit is flipped, otherwise it is left alone[6]. The Toffoli gate is used mainly in quantum algorithms and quantum error correction. For the readers we provided the truth values of the Toffoli gate shown in Table 2.2. Its matrix and circuit representation is (the two top lines represent the control qubit, the bottom line the target qubit)

$$\text{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.24)$$



We can apply our knowledge from the previous sections and show how we can build the Toffoli gate using only a set of single qubit gates. This process is shown in Figure 2.5. It can be shown that any unitary operation can be constructed to an arbitrarily good approximation just from the CNOT, Hadamard, phase and $\pi/8$ gates [8].

These two important sections about single and multiple qubit gates are important for further understanding quantum computation. We will be able to build quantum circuits which will be able to perform computation. In this chapter we used sources [4, 8, 9, 10].

[6] The Toffoli gate has been successfully realized in January 2009 at the University of Innsbruck, Austria.

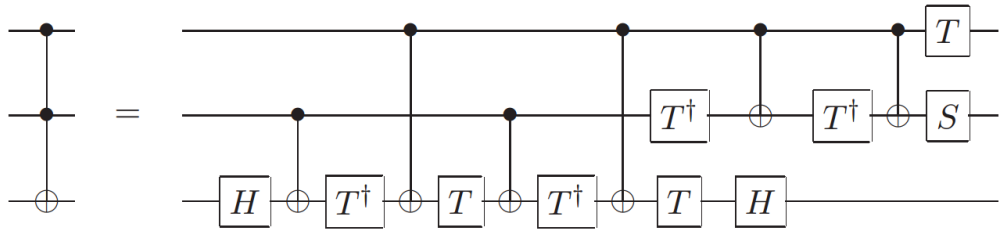


Figure 2.5: Implementation of the Toffoli gate using Hadamard, phase, CNOT and $\pi/8$ gate [8].

Table 2.2: Truth table of the Toffoli and Fredkin gates. For the Toffoli gate x and y are the control qubits and z is the target qubit before applying the Toffoli gate and f after applying the Toffoli gate. For the Fredkin gate x is the control qubit, y_1 and y_2 are the target qubits before applying the Fredkin gate and z_1 and z_2 are qubits after applying Fredkin gate.

Toffoli gate						Fredkin gate					
Input			Output			Input			Output		
x	y	z	x	y_1	f	x	y_1	y_2	x	z_1	z_2
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0	1
0	1	0	0	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	1	1	0	1	1
1	0	0	1	0	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0	1	1	1	0
1	1	0	1	1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1	1	1	1	1

Chapter 3

Quantum algorithms

Now it is time to compare classical and quantum computers in terms of algorithms. Are quantum algorithms more efficient then their classical counterparts? Is it possible to simulate a classical logic circuit using a quantum circuit? It would be surprising if the answer wouldn't be yes! In this chapter we will show a couple of examples of quantum algorithms (*Deutsch's alogrithm, quantum Fourier transform*) and introduce a strange but a astonishing behaviour of quantum computers called *quantum parallelism*. We will apply our quantum gates introduced in the previous chapter and show the advantages and power of quantum computing.

3.1 Quantum parallelism

The key for understanding quantum parallelism is superposition (Equation (2.1)) thanks to which we are able to evaluate a function $f(x)$ for many different values x simultaneously.

Let us consider a function $f(x) : \{0, 1\} \to \{0, 1\}$. Now we will construct a two qubit gate which will transform an arbitrary state $|x\rangle|y\rangle$ in the following way

$$|x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle. \tag{3.1}$$

Symbol $\oplus$ stands for addition modulo 2. This gate has two input qubits ($|x\rangle|y\rangle$) and two output qubits. We will call this gate U_f and it is depicted in Figure 3.1. It can be shown, that this gate is unitary. For better understanding we will assume $|y\rangle = |0\rangle$ and that $|x\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. We have already shown that this state can be produced by applying the Hadamard gate on state $|0\rangle$.

As shown in Figure 3.1 this gate returns the same $|x\rangle$ but more importantly the second qubit returns

$$|y\rangle = |0\rangle \to \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}, \tag{3.2}$$

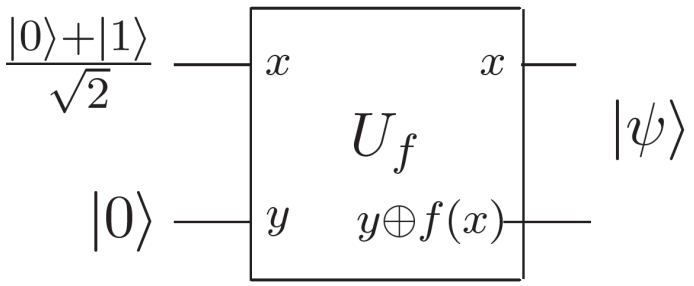


Figure 3.1: Two qubit U_f gate, where $|x\rangle$ is called the *data* register and $|y\rangle$ is called the *target* register [8].

which contains both $f(0)$ and $f(1)$ after a single run of the gate! This property of quantum computers is called *quantum parallelism*. Imagine a classical computer would have to evaluate multiple functions on multiple circuits, its quantum counterpart needs only one circuit for this task!

We can generalize this procedure on n number of bits. Now, let us consider a function $f(x) : \{0,1\}^n \to \{0,1\}$. We will start by acting with n Hadamard gates parallel on n qubits[1] in state $|0\rangle$ (we will denote n qubits in an arbitrary state x as $|x\rangle_n$). We will get state

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle. \tag{3.3}$$

Gate U_f will act on a $n+1$ size quantum register, which will change state of the input qubits similary as in Equation (3.1)

$$|x\rangle_n|y\rangle \to |x\rangle_n|y \oplus f(x)\rangle. \tag{3.4}$$

The output of the generalized U_f gate (we denote $\{0,1\}^n \equiv \{0,1,\dots 2^n-1\}$)

$$\begin{aligned} U_f \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0 \oplus f(x)\rangle \\ = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle. \end{aligned} \tag{3.5}$$

We have evaluated $f(x)$ for all x in a single step independently from the size of n! Now, as we mentioned in Section 2.1.2 the case of $n = 500$ qubits forming a quantum register contains more numbers than the number of atoms in the universe. We built a gate that evaluates $f(x)$ for all of these numbers in a single step! Unfortunately things aren't as good as they look. By measuring

[1] This operation is also called *Walsh–Hadamard transform*.

the quantum register $\sum_x |x\rangle|f(x)\rangle$ we would obtain only $f(x)$ for a single value x. Why would we evaluate functions on quantum computers if we could do the same thing on a classical computer? In the next section we will show that the strength is to extract information from the whole superposition of states $\sum_x |x\rangle|f(x)\rangle$.

3.2 Deutsch's algorithm

We are going to introduce our first quantum algorithm. The Deutsch algorithm is a simple algorithm based on the Quantum Fourier Transform which will be defined in the following chapter. Since this algorithm is very simple and easy to understand, it is ideal to demonstrate the key ideas of *quantum parallelism* and a property of quantum mechanics called *quantum interference*.

Let us again consider the one-bit function $f : \{0,1\} \to \{0,1\}$. Our problem is to determine the value $f(0) \oplus f(1)$. If $f(0) \oplus f(1) = 0$, then $f(0) = f(1)$ (f is a constant function without knowing the values of $f(0)$ and $f(1)$). If $f(0) \oplus f(1) = 1$, then $f(0) \neq f(1)$(we say that the function is *balanced*). The Deutsch algorithm is implemented by the circuit shown in Figure 3.2

The Deutsch Problem

Input: A black box for computing an unknown function $f : \{0,1\} \to \{0,1\}$.
Problem: Determine the value of $f(0) \oplus f(1)$ by making queries to f.

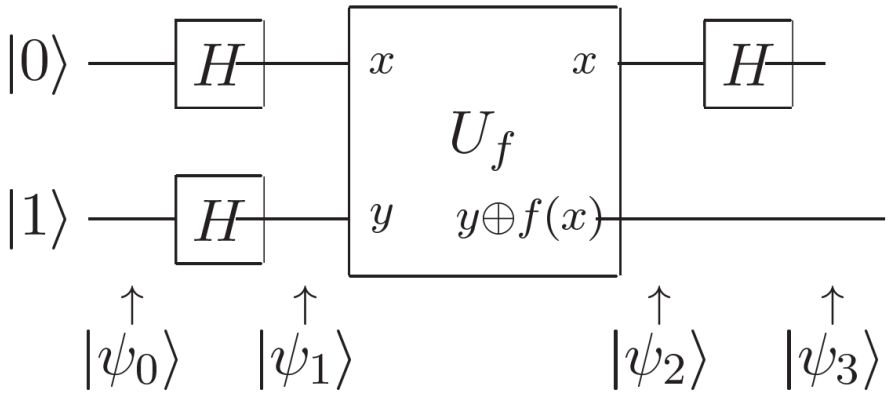


Figure 3.2: The quantum circuit representing Deutsch's algorithm [8]

Now, we prepare the first qubit as the superposition $(|0\rangle + |1\rangle)/\sqrt{2}$, but we will prepare the second qubit y as the superposition $(|0\rangle - |1\rangle)/\sqrt{2}$ (we can prepare such a state by applying the Hadamard gate on $|1\rangle$). We will show how each stage of the circuit will act on two qubits by following Figure 3.2.

First, the input state is

$$\psi_0 = |01\rangle. \tag{3.6}$$

After applying the Hadamard gate on both qubits we get

$$\psi_1 = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \equiv |+\rangle|-\rangle = \frac{1}{\sqrt{2}}|0\rangle|-\rangle + \frac{1}{\sqrt{2}}|1\rangle|-\rangle, \tag{3.7}$$

Let us show that by applying gate U_f represented by Equation (3.1) on state $1/\sqrt{2}(|0\rangle - |1\rangle)$ we get

$$\begin{aligned} f(x) = 0: &\quad \frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle \\ f(x) = 1: &\quad \frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = -|-\rangle. \end{aligned}$$

These two expressions differ by the factor (-1) which depends on the value of $f(x)$. We can rewrite both expressions in a more convenient way

$$\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = (-1)^{f(x)}\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} = (-1)^{f(x)}|-\rangle. \tag{3.8}$$

We apply gate U_f to ψ_1 and we get ψ_2

$$\begin{aligned} |\psi_2\rangle = \frac{(-1)^{f(0)}}{\sqrt{2}}|0\rangle|-\rangle + \frac{(1)^{f(0)}}{\sqrt{2}}|1\rangle|-\rangle &= \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|-\rangle \\ &= (-1)^{f(0)}\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}|-\rangle, \end{aligned} \tag{3.9}$$

where we used the fact, that $(-1)^{f(0)}(-1)^{f(1)} = (-1)^{f(0)\oplus f(1)}$. We get two different results of ψ_2 whether function f is constant or balanced

$$|\psi_2\rangle = \begin{cases} (-1)^{f(0)}|+\rangle|-\rangle & \text{if } f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|-\rangle & \text{if } f(0) \oplus f(1) \neq 0 \end{cases} \tag{3.10}$$

Finally we complete our algorithm by acting on the first qubit by the Hadamard gate

$$|\psi_3\rangle = \begin{cases} (-1)^{f(0)}|0\rangle|-\rangle & \text{if } f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|1\rangle|-\rangle & \text{if } f(0) \oplus f(1) \neq 0 \end{cases} \tag{3.11}$$

Now, by measuring the first qubit we may determine the value $f(0) \oplus f(1)$ and thus whether function f is constant or balanced. Again by comparing a quantum computer with a classical computer we see that in quantum computers

there is a connection between values $f(0)$ and $f(1)$. On the other hand, no connection can be found by evaluating these functions on a classical computer. We say that values $f(0)$ and $f(1)$ interfere. A generalization of the Deutsch algorithm called the *Deutsch-Jozsa algorithm* and can be found in [9, 11].

3.3 Quantum Fourier transform

What kind of tasks can a quantum computer solve more efficiently than a classical computer?

1. Algorithms based on the Fourier transform (for example the Deutsch algorithm is an algorithm based on the Fourier transform)
2. *Quantum search algorithms*.

The Fourier transform is an extremely important mathematical operation in almost every field of science. There are different kinds of Fourier transform, we will use the *discrete Fourier transform*. Let us assume a vector $\mathbf{x} = [x_0, x_1 \dots x_{N-1}]^T$, where $x_i \in \mathbb{C}$. The discrete Fourier transform of $\mathbf{x}$ is denoted by $\mathbf{y}$=DFT($\mathbf{x}$) where the Fourier coefficients of $\mathbf{y}$ are defined as

$$y_k \equiv \frac{1}{\sqrt{n}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k/N}. \tag{3.12}$$

The *quantum Fourier transform* (QFT) on an orthonormal basis $|0\rangle, |1\rangle \dots, |N-1\rangle$ is defined as

$$|j\rangle \to \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle. \tag{3.13}$$

Let us show the circuit representation of the QFT from which we will prove that QFT is unitary[2]. We will consider n qubit quantum computer which will apply the QFT on every qubit. According to [8] (with proof) the QFT can be given to following *product representation*

$$|j_1, j_2 \dots, j_n\rangle \to \frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\dots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right)}{2^{n/2}}, \tag{3.14}$$

where $0.j_l j_{l+1} \dots j_m$ represents the *binary fraction* $j_l/2 + j_l/4 + \dots j_m/2^{m-l+1}$.

[2]Remember, quantum algorithms represented by a quantum gate have to be unitary!

Building a circuit for the QFT

Input: Set of n qubits, Hadamard and modified phase shift gate
Output: Verifying Equation (3.14)

We will only need two gates, the Hadamard gate and a slightly modified phase shift gate which we will denote as R_k

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}. \tag{3.15}$$

However in our QFT circuit we will use it as a controlled operation (controlled-phase shift gate). Let us follow the circuit shown in Figure 3.3. As input we have n qubits in state $|j_1, j_2 \dots j_n\rangle$. First we act by the Hadamard gate on the first qubit and we get the state

$$\frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right) |j_2 \dots j_n\rangle. \tag{3.16}$$

Now we apply the controlled-R_2 on the first qubit and we get the state

$$\frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right) |j_2 \dots j_n\rangle. \tag{3.17}$$

We do this operation until $R_k = R_n$. We get

$$\frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right) |j_2 \dots j_n\rangle. \tag{3.18}$$

According to Figure 3.3 we apply a similar procedure on the second qubit. After applying the Hadamard gate and the controlled-R_2 through R_{n-1} on it we get

$$\frac{1}{2^{2/2}} \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_2 \dots j_n}|1\rangle\right) |j_3 \dots j_n\rangle. \tag{3.19}$$

Finally operations on the remaining qubits in an analogous way will give us

$$\frac{1}{2^{2/n}} \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_2 \dots j_n}|1\rangle\right) \dots \left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right). \tag{3.20}$$

By using swap operations we get the wanted Equation (3.14)

$$\frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_n j_{n-1}}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right)}{2^{n/2}}. \tag{3.21}$$

By deriving this long equation we have proven that the QFT is unitary since we only used unitary gates. The circuit that represents the QFT uses

in total $n(n+1)/2$ Hadamard and controlled-R_k gates. It can be shown that maximum $n/2$ swap gates are required. Therefore the difficulty of QFT circuit is $\Theta(n^2)$. In comparison with a classical computer, which computes the DFT using $\Theta(n2^n)$ gates we see that it needs exponentially more gates to perform the same operation.

Unfortunately we are not able to determine the amplitudes (due to wave-function reduction) but an important task that QFT enables is *phase estimation* which we will introduce in the next section.

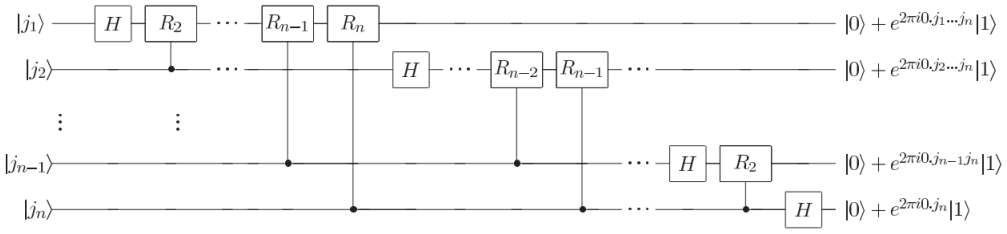


Figure 3.3: Quantum circuit implementing QFT [8].

3.3.1 Phase estimation

Phase estimation is an important part of quantum algorithms such as factorization. Let us consider an unitary operator U (we do not know its exact form) with eigenvector $|u\rangle$ and eigenvalue $e^{2\pi i\varphi}$. That is

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle. \tag{3.22}$$

We will use the QFT to an n bit estimation of the phase φ. To estimate the phase φ we will need two quantum registers. Again, by following the circuit in Figure 3.4 we will approach this problem.

Phase estimation problem

Input: Two quantum registers, Hadamard and controlled-U gate, eigenvalue $e^{2\pi i\varphi}$ and eigenvector $|u\rangle$ of gate U, inverse QFT
Output: Phase φ

The first quantum register contains t qubits in state $|0\rangle$. The second register will be prepared in state $|u\rangle$. First we are going to apply the Hadamard gate on the first register. After that, we apply controlled-U operation with the successive powers of two on the second register (for better understanding see Figure 3.4). The state of the first register will be

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right) = \frac{1}{2^{t/2}}\sum_{l=0}^{2^t-1} e^{2\pi i\varphi l}|l\rangle. \tag{3.23}$$

Due to Equation (3.22) the second register stays unchanged. In the next step we apply on the first register the *inverse* QFT which we can obtain by reversing the algorithm shown in the previous section.

The last step is to measure the first register and we get

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1} e^{2\pi i\varphi j}|j\rangle|u\rangle \rightarrow |\tilde{\varphi}\rangle|u\rangle \tag{3.24}$$

where $|\tilde{\varphi}\rangle$ is a good estimation of the phase $|\varphi\rangle$. We obtain the exact result of φ, if φ can be written exactly with a t bit binary expansion. In general the circuit shown in Figure 3.4 provides a good approximation of φ.

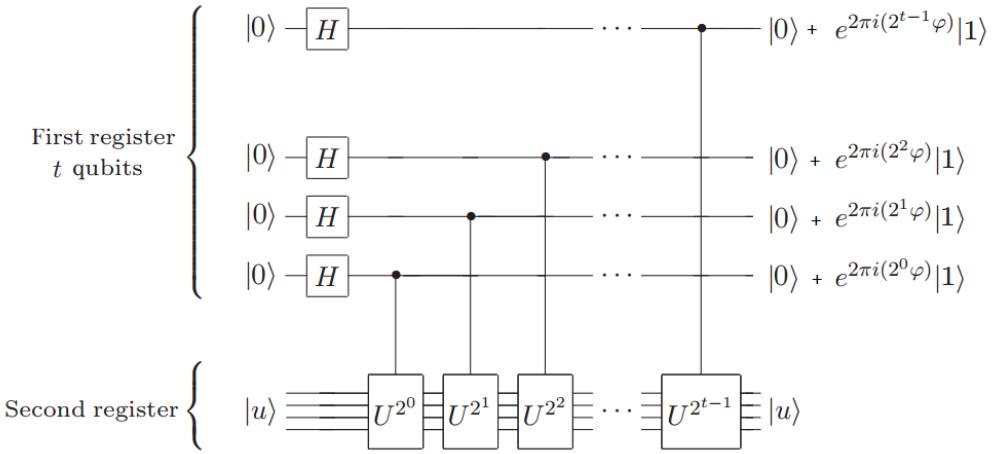


Figure 3.4: Quantum circuit implementing the phase estimation [8].

3.3.2 Order-finding

Now we will show the application of the phase estimation procedure on another procedure called order-finding. Let us assume two positive integers x and N, where $x < N$. Then we say the *order* of x in modulo N sense is defined as the least natural number r such that

$$x^r \bmod N = 1. \tag{3.25}$$

In other words we will want to find the period of the function $f(p) = x^p \bmod N$ since

$$f(p+r) = x^{p+r} \bmod N = [(x^p \bmod N) \cdot (\underbrace{x^r \bmod N}_{=1})] \bmod N = f(p). \tag{3.26}$$

Our task will be to find r for fixed parameters x and N. Quantum order-finding algorithm is nothing else but the phase estimation algorithm applied to the unitary operator

$$U|y\rangle = |xy \,(\bmod N)\rangle \tag{3.27}$$

It can be shown that the eigenstates of U for $0 \leq s \leq r-1$ are

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] |x^k \bmod N\rangle, \tag{3.28}$$

with eigenvalues $\exp\left[\frac{2\pi i s}{r}\right]$ from which thanks to the phase estimation procedure we can get the value of r. To run this procedure we need to implement a controlled-U^{2^j} which can be done by an algorithm called *modular exponentiation* [8]. More importantly we will need to prepare an eigenstate $|u_s\rangle$. Due to Equation (3.28) we would need r. Fortunately we can escape from this awkward situation by realizing that $|1\rangle$ is equal the superposition of eigenvectors

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle. \tag{3.29}$$

We emphasize that order-finding is believed to be a NP problem on a classical computer. NP problem means, that there is no known algorithm which could solve the problem in a *polynomial* (feasible) amount of time, but we are able to check the solution of the problem in a polynomial amount of time. For example the well-known *travelling salesman problem* is a NP problem. On the other hand a quantum computer can solve this problem efficiently, that is in a polynomial amount of time.

3.3.3 Factoring

Factoring is an essential part of cryptography. The well-known RSA[3] cryptosystem works on the principle of factoring

Let us consider a composite number N. Our goal will be to find a non-trivial factor of N. Factoring uses the order-finding algorithm shown in Section 3.3.2. This algorithm contains five simple steps

[3] Named after the last names of the authors: R. Rivest, A. Shamir and L. Adleman.

Factoring problem

Input: Composite number N.
Output: A non-trivial factor of N.

1. If N is even, return the factor 2.
2. Determine whether $N = a^b$ for integers $a \geq 1$ and $b \geq 2$, and if so return the factor a.
3. Randomly choose x in the range 1 to $N-1$. If $\gcd(x, N) > 1$ (gcd stands for greatest common divisor) then return the factor $\gcd(x, N)$.
4. Use the order-finding subroutine to find the order r of x modulo N.
5. If r is even and $x^{r/2} \neq -1 (\mathrm{mod}\ N)$ then compute $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$, and test to see if one of these is a non-trivial factor, returning the factor if so. Otherwise the algorithm fails.

Imagine if we could build a working quantum computer and implement this algorithm on it. This process requires $O((\log N)^3)$ operations (we get the exact result with a limited probability, but it is easy to verify if it divides N) thus this process is efficient and the quantum computer would be able to break the RSA cryptosystem[4]. On the other hand a classical computer needs $O\left(\exp\left[\left(\frac{64}{9}k\right)^{1/3}(\log k)^{2/3}\right]\right)$ where k is the number of bits of the factorized number. For big k it is almost impossible to break RSA.

This brings us to a question: By developing a quantum computer will our passwords or bank accounts be safe? This problem will require the development of a new area of quantum information called quantum cryptography.

In this chapter we used mainly sources [8, 9, 10].

[4]The RSA cryptosystem is the most commonly used system to encrypt 'secret' data.

Chapter 4

Latest development of quantum computers

There have been several attempts to construct a quantum computer on various universities. These computers work on different principles where qubits can be represented for example by two different polarizations of a photon or in the atom model the electron can exist in the 'ground' state $|0\rangle$ or in 'excited' state $|1\rangle$. Many examples are shown in book [11].

We will focus on the most discussed quantum computer developed by the private company called *D-Wave*. They announced their first quantum computer in 2007, called *Orion*. The first notable quantum computer was the D-Wave One on which scientific experiments were made and we will make an in depth view of this computer. Table 4.1 shows how the D-Wave computers have evolved in time. There has been a lot of discussion whether the 10 000 000$ machine really takes advantage of quantum mechanics and if it outperforms classical computers. In the following sections we will try to explain on what principle does D-Wave constructs their computers.

Table 4.1: Advancement of the D-Wave quantum computer

Name	Date of introduction	Number of qubits
Orion prototype	13-Feb-07	16
D-Wave One code named *Rainier*	11-May-11	128
D-Wave Two code named *Vesuvius*	early 2012	512

Figure 4.1: The logo of the D-Wave quantum computing company [12].

Since the D-Wave One has been on the market longer than the more powerful D-Wave Two and more experiments and scientific papers were written about D-Wave One, we will try to explain the fundamental basics on this computer.

As we know the most important part of a computer is its processor. The D-Wave processor is designed to harness a fundamental principle of nature that operate in both quantum and classical regimes – the propensity for all physical systems to minimize their free energy. The free energy minimalization in a classical system is often referred to as annealing. The D-Wave One is an *adiabatic quantum annealer* with up to 128 superconducting flux qubits. In the following sections we will explain what this means.

4.1 Superconducting flux qubits

As we mentioned earlier a qubit can be physically represented for example by the polarization of a photon. The D-Wave machine uses micrometer sized loops of superconducting metal interrupted by a number of Josephson junctions (usually one loop contains three junctions) [13] .

Josephson effect

Josephson effect describes the emergence of electric current between two superconductors separated by a thin layer of insulating material. It is a special case of quantum tunneling where particles pass through a seemingly impenetrable barrier. Devices using the Josephson effect can be in the form of microscopic electronic components called *Josephson junctions* [14]. Schematically it is shown in Figure 4.2 .

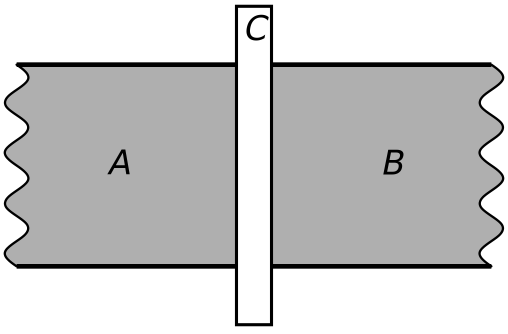


Figure 4.2: Diagram of a single Josephson junction. A and B represent superconductors, and C the weak link between them [14].

The computational basis states differ in having opposite flux (and circulating currents). For example if the current circulates clockwise we say, that the qubit

is in state $|0\rangle$ and if anti-clockwise, the qubit is in state $|1\rangle$. Now, in Section 2.1.1 we stated, that a qubit can exist in a superposition of states $|0\rangle$ and $|1\rangle$. What does it mean for our representation of a qubit? Thanks to the Josephson junctions in the loop the currents can 'flow both ways'. In Figure 4.3 we show schematically a diagram of energy versus applied flux.

According to the authors of article [13], the energy levels of state $|0\rangle$ (black line) and state $|1\rangle$ (red line) are shown near the applied magnetic field of $0.5\phi_0$ in the qubit loop. Next it can be seen, that the slope of E versus magnetic field is the circulating current. That means that these two classical states have opposite circulating currents. However, quantum mechanically, the charging energy couples these two states and results in an energy level repulsion at $\phi_{ext} = 0.5\phi_0$, so that there the system is in a linear superposition of the currents flowing in opposite directions. As the applied field is changed from below $\phi_{ext} = 0.5\phi_0$ to above, the circulating current goes from negative to zero at $\phi_{ext} = 0.5\phi_0$, to positive as shown in the lower graph of Figure 4.3 .

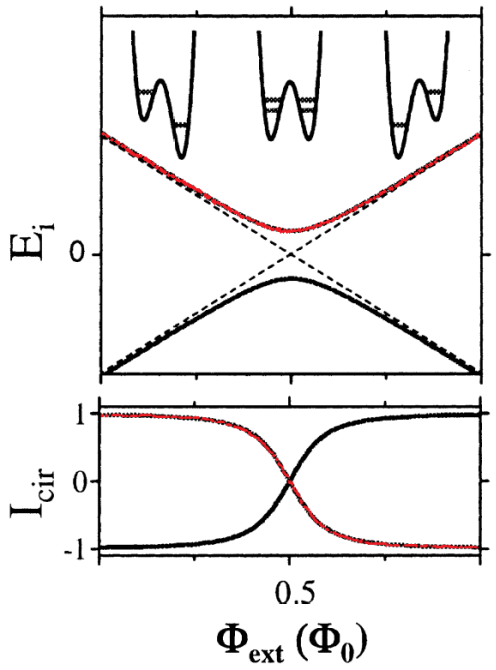


Figure 4.3: The energy levels for the basis state $|0\rangle$ (red line) and state $|1\rangle$ (black line) versus applied flux. The double well potentials are shown schematically above. On the lower graph we can see the circulating current in the qubit for both states as a function of applied flux. The units of flux are given in terms of the flux quantum [13].

4.2 Quantum annealing and adiabatic quantum computation

In the beginning of this chapter we stated, that D-Wave One is an adiabatic quantum annealer. In the following sentences we will try to explain what it means.

4.2.1 Annealing

Annealer refers to *quantum annealing* (QA) [15, 16]. First we will have to distinguish simulated annealing and quantum annealing. A nice example from metallurgy may clarify what annealing means. Annealing a metal involves heating it and then cooling it. Before heating, the metal is filled with defects (metastable 'high energy' state). After heating and cooling it, the metal becomes crystalline and defect-free (the minimum free energy).

Simulating this type of thermal annealing on a classical computer is known as simulated annealing (SA). Instead of having a fixed landscape through which to anneal (as in the metallurgical example), in simulated annealing a programmer defines what the energy landscape is. This energy landscape is crafted so that its global minimum is the answer to the problem to be solved, and low-lying local minima are good approximations.

Minimizing free energy in a quantum systems is called *quantum annealing*. Similary to classical annealing, all quantum systems are driven to minimize their free energy. In non-programmable scenarios (metal annealing), it has been shown that quantum annealing can hasten the energy minimisation process.

This method is particularly useful for problem where the search space is discrete with many local minima, for example finding the *ground state of a spin glass using quantum tunneling* (in the following section we will explain what it means). That is the reason why D-Wave hopes their that quantum processor will be faster than its classical counterpart.

Thus, D-Wave processors take advantage of quantum annealing. QA processor can be operated as a universal quantum computer. In this regime of operation, the computational model is referred to as *adiabatic quantum computation* (AQC). It can be thought as the long-time limit of QA. In the next section AQC is explained more precisely [17].

4.2.2 Adiabatic quantum computation

Adiabatic means that the computer relies on the adiabatic theorem, which says:

Theorem 2 (Adiabatic theorem). *A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.*

In other words, if we change slowly the external conditions of a quantum mechanical system, its functional form also adapts to this change. On the other hand if we change the external conditions quickly there is insufficient time for the functional form to adapt. Adiabatic quantum computation can be described in three simple steps [18]:

1. We find a Hamiltonian whose ground state describes the solution to the problem of interest.
2. A system with a simple Hamiltonian is prepared and initialized to the ground state.
3. The simple Hamiltonian is adiabatically evolved to the complex Hamiltonian.

Thanks to the adiabatic theorem the whole new system stays in the ground state and at the end the state of the whole system describes the solution of the problem.

4.3 Experiment on the D-Wave One computer

D-Wave One are analog embodiments of the optimization version of the Ising spin glass model in a magnetic field problem [19]. The processor is designed to hasten convergence of the energy of the system towards the ground state energy. If the system is able to reach its ground state, the configuration of variables returned is the exact solution of the problem. If it is able to reach only a low-lying local minimum, the configuration of variables returned is an approximate solution [17].

The D-Wave tries to find the ground state of a spin glass model with the Hamiltonian

$$H = -\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z, \tag{4.1}$$

where σ_i^z is the Pauli matrix for qubit i, h_i is the local bias on qubit i and J_{ij} is the coupling strength between qubits i and j.

A problem instance is encoded in the h and J values, which are user-programmable!

The scientists, who published article [20] made the following three experiments:

1. Performed quantum annealing on the D-Wave One (DW) device.
2. Performed simulated quantum annealing[1] (SQA).

[1] It is also possible to construct a simulated quantum annealing algorithm. Due to its complexity, we refer the reader to [20].

3. Simulated classical annealing (SA).

Experimental setup

They performed this experiment with 1000 different random couplings $J_{ij} = \pm 1$ (and some of the data also random fields $h_i = \pm 1$) in Equation (4.1) and for every single input they performed $M = 1000$ annealing runs and determined whether the system reached the ground state. Then they compared the experimental results gained from DW to SQA and SA. Mainly, they compared the distribution of the success probabilities and the correlation between the D-Wave device and the other models. Results are shown in Figure 4.4

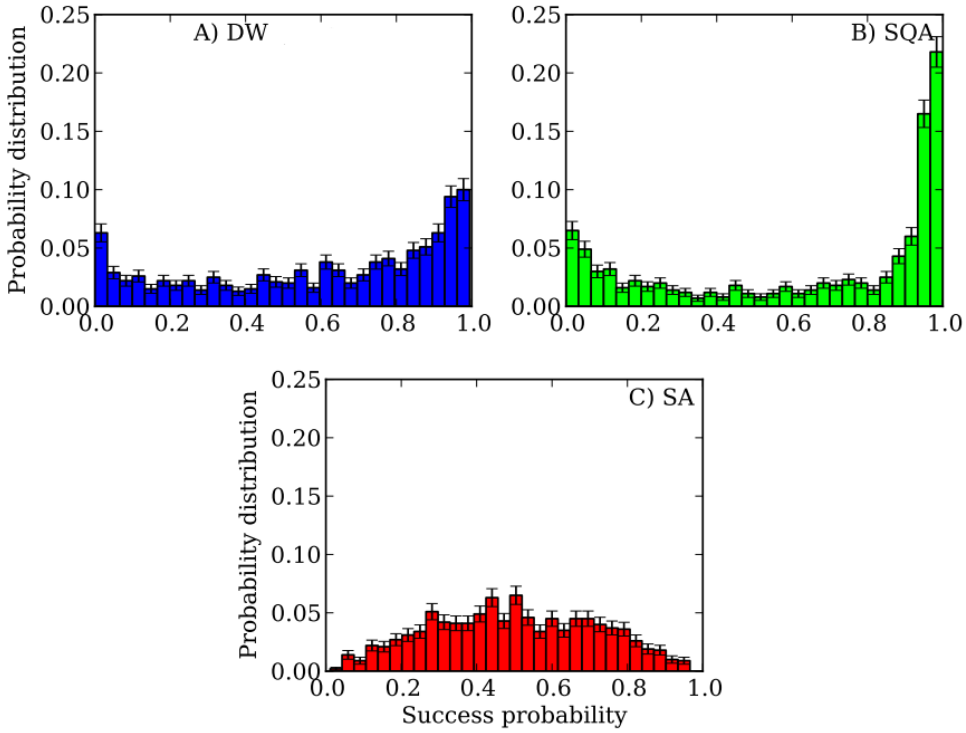


Figure 4.4: Succes probabability distributions for different experiments. We can see similar bimodal distributions for the D-Wave results (DW, panel A) and the simulated quantum annealer (SQA, panel B), and unimodal distribution for the simulated annealer (SA, panel C) [20].

Conclusion

In the first test they counted for each instance the number of runs M_{GS} in which the ground state was reached to determine the success probability as $s = M_{GS}/M$. In Figure 4.4 we see that DW results match well with SQA but poorly with SA. There has been a lot of discussion about how 'quantum' the D-Wave machine is [21]. The authors of article [20] claim, that quantum annealing with more than one hundred qubits takes place in the D-Wave One device. The key evidence is the correlation between the success probabilities on the

device and a simulated quantum annealer (see Figure 4.4 and the similarity of graphs on panel A and B).

The authors also compared the computational time of these algorithms. They stated, by considering the pure annealing time, the performance matches that of a highly optimised classical annealing code on a high-end CPU.

Figure 4.5: Three D-Wave 'boxes' side by side [12].

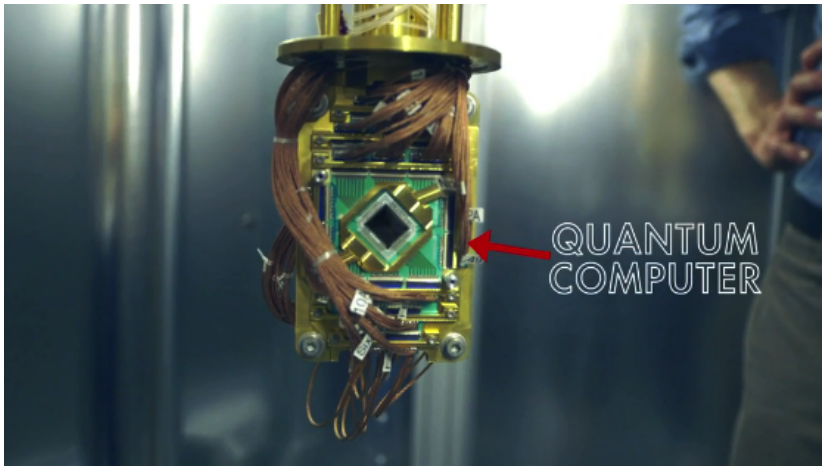


Figure 4.6: The inside of the D-Wave 'box'. The heart of the D-Wave quantum computer – its quantum processor [12].

Summary and Conclusion

In this work we introduced the basic principles of quantum computers. Our first goal was to introduce the fundamentals of quantum computing. In Chapter 2 we have shown basic quantum gates and their geometrical meaning on the Bloch sphere. These gates are an essential part of building more complicated quantum circuits performing different calculations.

In Chapter 3 we introduced quantum algorithms. Clearly there are many more algorithms, such as *Grover's search algorithm* (see [9]), but our main goal was to introduce an important property of quantum computers called quantum parallelism. Understanding parallelism enabled us to build more complicated quantum algorithms such as Deutsch's algorithm. By presenting the quantum Fourier transform we were able to derive a quantum algorithm for factoring which is extremely useful in cryptography.

Our second task in this thesis was to summarize the present development of quantum computers from scientific papers. At several universities and laboratories there are attempts to build quantum computers however we have chosen to examine the Canadian D-Wave One device. We explained the principles how it works and presented an experiment (finding the ground state of the Ising spin glass model) comparing the properties of D-Wave computer with that of a classical one. It is worth saying that due to noisy, high error-rate qubits it is not possible to run the previously mentioned algorithms (for example factorization) on the D-Wave One device.

However as stated in article [20] it should be interesting to adress the open question of quantum speedup on future devices with more qubits. Going to even larger problem sizes we soon approach the limits of classical computers. A quantum annealer showing better scaling than classical algorithms for larger problem sizes would be a big breakthrough, validating the potential of quantum information processing to outperform its classical counterpart.

Bibliography

[1] A. Ekert, P. Hayden, and H. Inamori. *Basic concepts in quantum computation*. arXiv: 0011013v1, 2000.

[2] J. Sibik. *Kvantové výpočty*. Prague, 2008. Bachelor thesis, Charles University in Prague.

[3] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Marianton, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, A. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M Martinis. *Computing prime factors with a Josephson phase qubit quantum processor*. arXiv:1202.5707, 2012.

[4] Wikipedia, The Free Encyclopedia. *Quantum gates*. [Online; accessed 27-April-2014]. Available from: `http://en.wikipedia.org/wiki/Quantum_gate`.

[5] G. J. Milburn. Quantum Optical Fredkin Gate. *Physical Review Letters*, 62:2124–2127, 1989.

[6] L. Skála. *Úvod do kvantové mechaniky*. Karolinum, Prague, 2012.

[7] Wikipedia, The Free Encyclopedia. *Moore's law*. [Online; accessed 30-April-2014]. Available from: `http://en.wikipedia.org/wiki/Moore's_law`.

[8] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge University Press, New York, 2010.

[9] S. Imre and F. Balázs. *Quantum Computing and Communications - An Engineering Approach*. First Edition. John Wiley & Sons, Ltd, Chichester, 2005.

[10] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. First Edition. Oxford University Press, New York, 2007.

[11] H. O. Everitt. *Quantum Computation and Quantum Information*. First Edition. Springer, New York, 2010.

[12] D-Wave Systems, Inc. [Online; accessed 15-April-2014]. Available from: http://www.dwavesys.com/.

[13] T. P. Orlando, S. Lloyd, L. S. Levitov, K. K. Berggren, M. J. Feldman, M. F. Bocko, J. E. Mooij, C. J. P. Harmans, and C. H. van der Wal. Flux-based superconducting qubits for quantum computation. *Physica C*, 372:194–200, 2002.

[14] Wikipedia, The Free Encyclopedia. *Josephson effect*. [Online; accessed 13-April-2014]. Available from: http://en.wikipedia.org/wiki/Josephson_effect.

[15] Wikipedia, The Free Encyclopedia. *Quantum annealing*. [Online; accessed 15-April-2014]. Available from: http://en.wikipedia.org/wiki/Quantum_annealing.

[16] A. Finnila, M. Gomez, M. Sebenik, C. Stenson, and J. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219:343–348, 1994.

[17] D-Wave systems, Inc. *An Introduction to Quantum Annealing*. [Online; accessed 29-April-2014]. Available from: http://dwave.files.wordpress.com/2007/08/20070810_d-wave_quantum_annealing.pdf.

[18] Wikipedia, The Free Encyclopedia. *Adiabatic quantum computation*. [Online; accessed 29-April-2014]. Available from: http://en.wikipedia.org/wiki/Adiabatic_quantum_computation.

[19] Wikipedia, The Free Encyclopedia. *Spin glass*. [Online; accessed 30-April-2014]. Available from: http://en.wikipedia.org/wiki/Spin_glass.

[20] S. Boixo, T. F Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer. *Quantum annealing with more than one hundred qubits*. arXiv: 1304.4595, 2013.

[21] S. W. Shin, G. Smith, J. A. Smolin, and U. Vazirani. *How "Quantum" is the D-Wave Machine?* arXiv:1401.7087, 2014.

List of Figures

1.1 CPU transistor counts against dates of introduction. Note the logarithmic vertical scale [7]. 3

2.1 Bloch sphere [8]. 6
2.2 Controlled U operation. 12
2.3 Controlled phase shift gate on the left side as a two qubit and on the right side as a single qubit operation [8]. 12
2.4 Controlled phase shift gate and equivalent circuit for two qubits [8]. 13
2.5 Implementation of the Toffoli gate using Hadamard, phase, CNOT and $\pi/8$ gate [8]. 15

3.1 Two qubit U_f gate, where $|x\rangle$ is called the *data* register and $|y\rangle$ is called the *target* register [8]. 17
3.2 The quantum circuit representing Deutsch's algorithm [8] 18
3.3 Quantum circuit implementing QFT [8]. 22
3.4 Quantum circuit implementing the phase estimation [8]. 23

4.1 The logo of the D-Wave quantum computing company [12]. . . 26
4.2 Diagram of a single Josephson junction. A and B represent superconductors, and C the weak link between them [14]. . . . 27
4.3 The energy levels for the basis state $|0\rangle$ (red line) and state $|1\rangle$ (black line) versus applied flux. The double well potentials are shown schematically above. On the lower graph we can see the circulating current in the qubit for both states as a function of applied flux. The units of flux are given in terms of the flux quantum [13]. 28
4.4 Succes probabability distributions for different experiments. We can see similar bimodal distributions for the D-Wave results (DW, panel A) and the simulated quantum annealer (SQA, panel B), and unimodal distribution for the simulated annealer (SA, panel C) [20]. 31
4.5 Three D-Wave 'boxes' side by side [12]. 32
4.6 The inside of the D-Wave 'box'. The heart of the D-Wave quantum computer – its quantum processor [12]. 32

List of Tables

2.1 Truth table of the CNOT gate ($\oplus$ stands for addition modulo two). 11
2.2 Truth table of the Toffoli and Fredkin gates. For the Toffoli gate x and y are the control qubits and z is the target qubit before applying the Toffoli gate and f after applying the Toffoli gate. For the Fredkin gate x is the control qubit, y_1 and y_2 are the target qubits before applying the Fredkin gate and z_1 and z_2 are qubits after applying Fredkin gate. 15

4.1 Advancement of the D-Wave quantum computer 26